

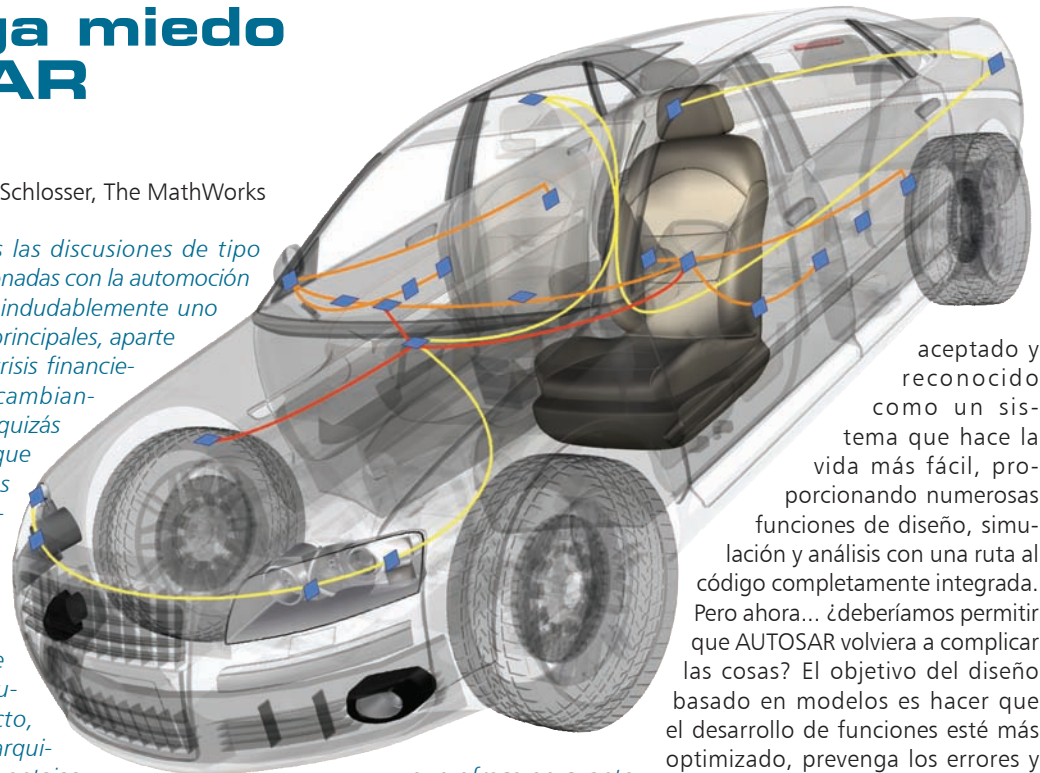
No le tenga miedo a AUTOSAR

Por Guido Sandmann y Dr. Joachim Schlosser, The MathWorks

 The MathWorks
www.mathworks.com

En casi todas las discusiones de tipo técnico relacionadas con la automoción AUTOSAR es indudablemente uno de los temas principales, aparte de la actual crisis financiera que está cambiando las reglas quizás incluso más que los requisitos generales sobre eficacia energética y protección medioambiental. Ya se ha escrito mucho al respecto, acerca de su arquitectura y las ventajas a reutilización, capacidad de intercambio y redistribución de funciones. Pero... ¿qué aporta AUTOSAR a quienes realmente se encargan de desarrollar el conjunto de funciones de un coche, es decir, los desarrolladores de funciones? Al hablar con ellos, muestran incertidumbre e incluso cierta resistencia a AUTOSAR; su miedo es obvio: ¡AUTOSAR podría incrementar la carga de trabajo y el esfuerzo de implementación en vez de reducirlos! La buena noticia es que muchas de estas dudas desaparecen en cuanto se conocen los datos reales.

Con este artículo queremos señalar algunos aspectos del desarrollo de componentes de software de AUTOSAR que no se suelen conocer bien. Describiremos por qué el diseño basado en modelos y AUTOSAR encajan a la perfección y por qué esta combinación ayuda a los ingenieros a disipar su escepticismo. El hecho más importante es: no hay necesidad de añadir bloques a un modelo para que sea compatible con AUTOSAR.



aceptado y reconocido como un sistema que hace la vida más fácil, proporcionando numerosas funciones de diseño, simulación y análisis con una ruta al código completamente integrada. Pero ahora... ¿deberíamos permitir que AUTOSAR volviera a complicar las cosas? El objetivo del diseño basado en modelos es hacer que el desarrollo de funciones esté más optimizado, prevenga los errores y encuentre antes el los que queden. El objetivo sigue siendo el mismo: ahorrar tiempo y dinero (y lo mismo se aplica a AUTOSAR). Los dos buscan la abstracción del hardware y la reutilización de componentes de un modo más fácil. Por esta razón, conseguir la igualdad de los objetivos era la máxima que The MathWorks tenía en mente cuando incorporó el concepto AUTOSAR a Simulink®.

La pregunta es: ¿qué nivel de conocimientos debe tener un ingeniero de desarrollo de funciones acerca

La confusión que existe con la meta-información de AUTOSAR

A pesar de que es cierto que el aumento de flexibilidad en términos de desacoplamiento de las aplicaciones del hardware de destino en una red de UCE conlleva ciertas consideraciones adicionales, sin duda esto no supondrá una tremenda carga para el desarrollador de funciones. El desarrollo de algoritmos que hagan que la conducción sea más segura, reduzcan el consumo de combustible o incluso controlen nuevas técnicas de actuación ya es bastante difícil. El diseño basado en modelos con Simulink® y Stateflow® está ampliamente

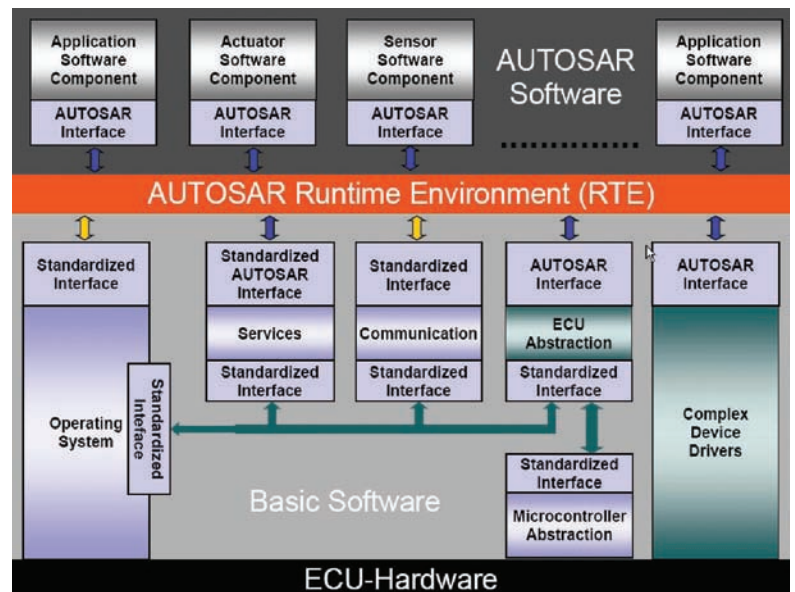


Figura 1. Arquitectura de software de AUTOSAR (Fürst, 2008)

de AUTOSAR? Aparte de asimilar la idea básica de una arquitectura de software en capas, ¿es realmente necesario adentrarse en las profundidades del entorno de tiempo de ejecución y el software básico? ¿Le ayuda esto a desarrollar sus productos con una calidad elevada? ¿Qué partes de la No le tenga miedo a AUTOSAR se deben conocer desde el punto de vista del desarrollo de funciones?

Resulta que todo lo que está por debajo del RTE apenas tiene interés para el aspecto funcional. Está claro que la comunicación debe producirse, pero ese tema ya está encapsulado a día de hoy. Está claro que los periféricos de la UCE deben dirigirse a través de controladores de dispositivos, pero eso también está encapsulado. Así pues, si ambos aspectos no son aplicables, el proceso de desarrollo se enfrentará a la necesidad de un cambio de todas formas, a fin de apartar esta complejidad del desarrollador de funciones.

La trampa del conjunto de bloques

¿Qué significa esto? En primer lugar, no hay necesidad de añadir bloques a un modelo para que sea compatible con AUTOSAR. El ingeniero de diseño deja intacta la estructura del modelo y se limita a aplicar un conjunto de configuraciones distinto. El concepto de conjunto de configuraciones de Simulink resulta esencial aquí, ya que es capaz de modificar los detalles de la implementación, aunque el algoritmo sigue siendo el mismo independientemente de si un puerto de entrada raíz es un puerto de Simulink normal, un receptor de AUTOSAR o un puerto para acceder de forma directa al software básico de AUTOSAR. Es simplemente un tema de configuración.

Así pues, no hay necesidad de modificar los modelos y usar algún tipo de conjunto de bloques de AUTOSAR. De hecho, no existe tal conjunto de bloques. Los clientes no pueden comprar a The MathWorks ningún paquete de AUTOSAR; no lo necesitan porque todo está incluido en forma de mejoras de las herramientas existentes. Aunque es habitual que en el desarrollo de

aplicaciones especiales exista algún conjunto de bloques adicional para aumentar la eficacia del modelado, no es necesario introducir ningún bloque nuevo para modelar un componente de software de AUTOSAR.

Los conjuntos de bloques de Simulink compendian funciones que son, por ejemplo, especiales para crear algoritmos de procesamiento de señales avanzado como filtros, códecs de comunicación o análisis espectral o procesamiento de vídeo e imagen. Otros contienen solucionadores y construcciones para modelar plantas (controladas) de acuerdo a sus principios físicos en vez de tener que preocuparse de las ecuaciones. Todas estas soluciones tienen en común que permiten modelar el comportamiento y las arquitecturas de un modo más fácil que si se emplean bloques básicos. El modelado físico, por ejemplo, añade la capacidad de disponer de conexiones de señales bidireccionales entre los bloques de construcción para transmitir la información en ambas direcciones, al contrario que ocurre con las líneas de señales dirigidas en el entorno de Simulink normal. Esto facilita el modelado de conexiones que extraen "fuerza" e "inercia" en una línea.

Por el contrario, las construcciones de modelado esenciales que se necesitan para desarrollar los componentes de software de AUTOSAR ya existen en Simulink y Stateflow, tanto si son puertos considerados como tales, ejecutables múltiples considerados como modelos de velocidades de muestreo múltiples o bien subsistemas de llamadas de funciones o tipos de datos complejos.

Sobrecarga en el tiempo de ejecución y tamaño del código

Una de las mentiras más famosas que existen con respecto a AUTOSAR es la opinión sobre la sobrecarga en la eficacia del tiempo de ejecución y el tamaño del código que provocaría. A menudo, tanto los ingenieros como los directores de proyectos temen que el código compatible con AUTOSAR que se genera (incluidos el RTE y el software básico) dé lugar a un código con un tamaño 2 o 3 veces superior y, al mismo

tiempo, una ralentización similar en cuanto a tiempo de ejecución y eficacia. La versión 2.1 de AUTOSAR fue la primera que se utilizó en serio para implementar e integrar aplicaciones compatibles con AUTOSAR. Muchos fabricantes OEM y proveedores cuentan ya con una biblioteca de "software básico" interna (a veces se denominan "Standard Cores"). Estas bibliotecas no son compatibles con el estándar de AUTOSAR pero han sido desarrolladas conforme al mismo objetivo: desacoplar las aplicaciones del hardware de destino. Los proyectos de AUTOSAR actuales a menudo se basan en versiones empaquetadas de estas bibliotecas que permiten la utilización y el desarrollo de los componentes de software de AUTOSAR.

Los fabricantes OEM y los proveedores ya están publicando sus primeras experiencias con proyectos de AUTOSAR y los resultados son muy prometedores (Lajtkep, 2008 y Nesci W., 2008). La tendencia muestra un aumento de consumo de código y memoria en este sector de hasta un 20% (no de un 200% o mayor). Por otra parte, se han publicado declaraciones confirmando que AUTOSAR es un buen sistema para reducir la complejidad. Estas excelentes experiencias constituyen un motivo suficiente para no tener miedo a AUTOSAR y aceptar las –tempranas– experiencias con el incremento de código y tamaño de memoria.

AUTOSAR y el diseño basado en modelos

Ahora que hemos dejado claro qué es lo que no debe preocupar demasiado a un desarrollador de funciones, deberíamos echar un vistazo a lo que constituyen las grandes ventajas del diseño basado en modelos cuando se combina con AUTOSAR. Es importante reconocer que no todo el que usa MATLAB y Simulink ya realiza trabajos de diseño basado en modelos, así que vamos a señalar algunas de las mejores prácticas de la adopción del diseño basado en modelos, ampliamente discutidas (Smith, Prabhu y Friedman, 2007), las cuales se aplican en particular a los ingenieros y el uso que hacen de sus herramientas.

Use los modelos para al menos dos cosas: "la regla de dos"

Con demasiada frecuencia, muchas organizaciones que hacen un gran trabajo de modelado sólo utilizan los modelos para un único objetivo. Hemos visto a clientes que usan los modelos únicamente para la generación de código, sin llegar a pulsar jamás el botón "Simular" en Simulink. A fin de obtener la máxima rentabilidad de la inversión realizada en software, es preciso utilizar la formación de las personas y los modelos de las organizaciones para al menos dos objetivos distintos. Además de la generación de código de producción, un objetivo podría ser la validación de requisitos a través de la simulación que se realiza en primer lugar, así como la generación automática de documentación. O también el uso de modelos para la creación de parámetros de controladores tomando como base la teoría de control lineal con Simulink Control Design para, a continuación, crear prototipos rápidamente, ya sea a través de xPC o bien con las soluciones de hardware de nuestro socio en Alemania, dSPACE.

Pregúntese a sí mismo: "¿qué hago con los modelos?" y "¿qué hacen mi equipo, mi departamento y mi organización con los modelos?". La regla de dos se aplica a todos los niveles.

Trate los modelos como única fuente de verdad

Esto nos lleva hasta la siguiente mejor práctica. Si encuentra un defecto de sistema en uno de los productos de trabajo de aguas abajo, no caiga en la tentación de solucionarlo allí mismo, ya que separará la especificación ejecutable de la realidad e impedirá su reutilización en el futuro. Arregle los defectos en el modelo. Su cadena de herramientas debería permitirle volver a descender rápidamente por la escala de abstracción mediante la automatización de los pasos de generación necesarios. Si cambia algo, hágalo en el modelo; así obtendrá todas las ventajas que ofrecen el análisis, la simulación y la implementación instantánea.

Céntrese en el diseño, no en la generación de código

El modelado permite pensar en el comportamiento del sistema en vez de en la implementación demasiado pronto. Pero con demasiada frecuencia las personas modelan pensando

en el código C que desean. Si empieza a hacer eso, ya no estará modelando, sino programando gráficamente y, en consecuencia, malgastando la energía del modelo. Si realmente quiere que un trozo de código aparezca en el resultado final, no vuelva a dibujarlo. Inclúyalo empleando las posibilidades de la integración de código heredado. Por lo demás, piense en el comportamiento, no en el código.

Asóciese con sus proveedores de herramientas

Las organizaciones inteligentes aprenden de sus propios errores. Las organizaciones realmente inteligentes aprenden de los errores de otros. Pero... ¿quién es capaz de recopilar esos errores y transformarlos en oportunidades de aprendizaje? Como los proveedores de herramientas están ampliamente involucrados en apoyar las transiciones al diseño basado en modelos, acumulan una gran experiencia que pueden usar para reducir de forma significativa la empinada cuesta de la proverbial curva de aprendizaje. Al hacerlo permiten que las organizaciones comprendan en un plazo más breve las ventajas que ofrece el diseño basado en modelos. De este modo se consigue alcanzar antes el umbral de rentabilidad de la inversión. Así pues, si se asocia con sus proveedores y permite que se involucren en su plan de transición, podrá sacar partido de sus experiencias, evitar dificultades frecuentes y llevar a cabo una transición perfecta.

Cómo encaja AUTOSAR en el diseño basado en modelos

El papel (y, al mismo tiempo, la importancia) del diseño basado en modelos dentro de los procesos de desarrollo tanto de los fabricantes de

automoción como de los proveedores ya se ha explicado arriba. La complejidad de las funciones, las interconexiones y las variantes de las UCE obligan a los ingenieros del sector de la automoción a cambiar de marcha a fin de mantenerse en la competición, dando total sentido al lema de AUTOSAR "Coopera en los estándares y compite en la implementación".

No sólo las simulaciones ofrecen una visión de los aspectos dinámicos y algorítmicos de un sistema, sino que los modelos también se usan habitualmente para diversas tareas constructivas, entre las que se incluyen:

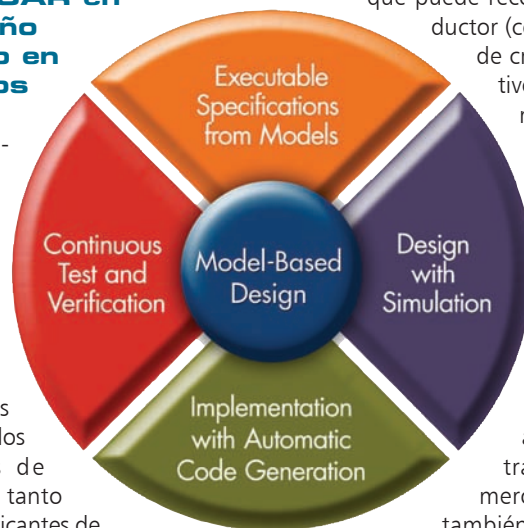
- Actuar como especificaciones ejecutables
- Comunicar los requisitos de componentes y las definiciones de interfaz entre el cliente y el proveedor
- Proporcionar prototipos virtuales de sistemas de vehículos y modelos de conductores, condiciones de carreteras y otras condiciones ambientales para el desarrollo de algoritmos
- Habilitar la generación de código automática de los algoritmos de software para los sistemas de producción

Estas actividades del diseño basado en modelos mantienen el proceso de ingeniería centrado en la prevención de errores y la detección temprana de errores. La verificación y la validación tempranas en un proyecto pueden reducir los riesgos asociados a una detección tardía de errores.

AUTOSAR deja bien claro que los ingenieros del sector de la automoción deben avanzar al siguiente nivel; la topología de las funciones que puede reconocer el conductor (como el control

de crucero adaptativo) no se detiene al llegar a las UCE. Una aplicación de este tipo se distribuye a través de diversas unidades de control dentro del vehículo y, aunque no se trata de un número fijo de UCE, también podría variar

Figura 2. Diseño basado en modelos



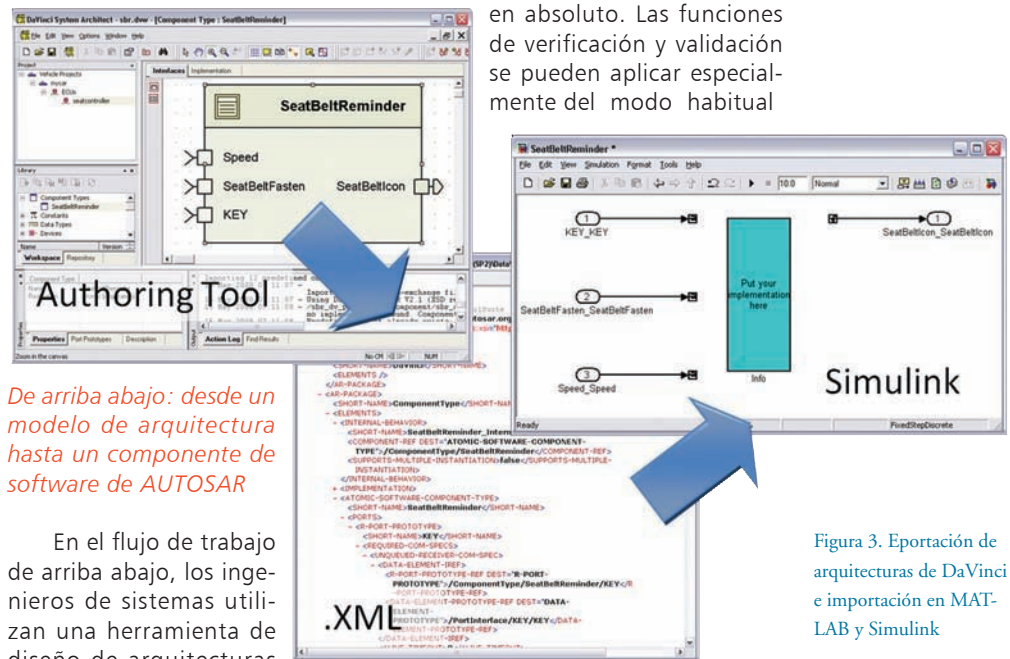
dependiendo del número de plataformas de vehículos. AUTOSAR ofrece un enfoque para introducir otro nivel de abstracción; la consideración a nivel de sistema pasa a un primer plano mientras que el enfoque relativo a los nodos (en términos de una sola UCE) pierde importancia.

El diseño basado en modelos es totalmente coherente con el enfoque de AUTOSAR. Una consideración de arriba abajo empieza desde una perspectiva de arquitectura de sistema. A fin de atender las notaciones específicas de AUTOSAR, se utiliza una herramienta de creación como DaVinci de Vector Informatik para describir adecuadamente la topología de una red de UCE. Los formatos de descripción definidos de AUTOSAR permiten una sencilla integración con herramientas adicionales para desarrollar el comportamiento funcional y los algoritmos de control, así como el software que implementa estas funciones. Los detalles de los posibles flujos de trabajo se describen en los próximos subapartados.

El éxito de AUTOSAR depende de la interoperabilidad de las herramientas a lo largo del Ciclo en forma de V. Las herramientas de creación deben trabajar con entornos de comportamiento funcional y generación de código, así como con entornos de configuración de software básico y generación de RTE.

Compatibilidad entre Simulink y Stateflow con Real-Time Workshop Embedded Coder y los importantes requisitos de funcionalidades de AUTOSAR

A fin de ilustrar algunos de los aspectos mencionados arriba, veamos cómo funcionan todos los elementos en conjunto. El enfoque de MathWorks basado en MATLAB®, Simulink®, Stateflow® y Real-Time Workshop® Embedded Coder para generar código compatible con AUTOSAR sigue un proceso transparente e intuitivo y es capaz de admitir dos flujos de trabajo distintos: de arriba abajo y de abajo arriba (Sandmann, Schlosser y Thompson, 2008). Puede ver una aplicación en el mundo real en (Köhler y Reck, 2007).



De arriba abajo: desde un modelo de arquitectura hasta un componente de software de AUTOSAR

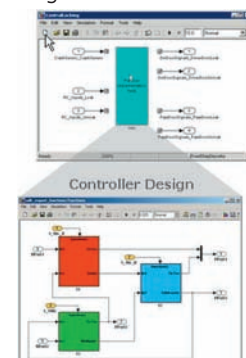
En el flujo de trabajo de arriba abajo, los ingenieros de sistemas utilizan una herramienta de diseño de arquitecturas (denominada "herramienta de creación" en la terminología de AUTOSAR), como el conjunto de herramientas DaVinci de Vector Informatik, para diseñar la arquitectura de la red de UCE de un vehículo. Por supuesto que esto no restringe la posibilidad para usar otras herramientas de creación. Seguidamente los ingenieros exportan una descripción XML de los componentes correspondientes: la descripción de componente de software de AUTOSAR. Este archivo contiene toda la información necesaria sobre los componentes, como runnables, descripciones de interfaz, tipos de datos, etc. Con la solución AUTOSAR de MathWorks, los ingenieros importan este archivo XML y generan de forma automática un modelo estructural de Simulink que contiene los bloques de interfaz (puertos de entrada y de salida) y la configuración relativa a AUTOSAR para estos objetos del modo definido. La Figura 2 ilustra este flujo de trabajo. A partir de este modelo estructural, los ingenieros de diseño desarrollan el modelo de controlador con Simulink, Stateflow y los conjuntos de bloques acompañantes, tal y como se muestra en la Figura 3.

La estructura de bloques del modelo se puede conservar y, gracias a las estables interfaces mencionadas anteriormente, los ingenieros no necesitan realizar cambios globales

en absoluto. Las funciones de verificación y validación se pueden aplicar especialmente del modo habitual

que aparece descrito en el apartado relativo al diseño basado en modelos.

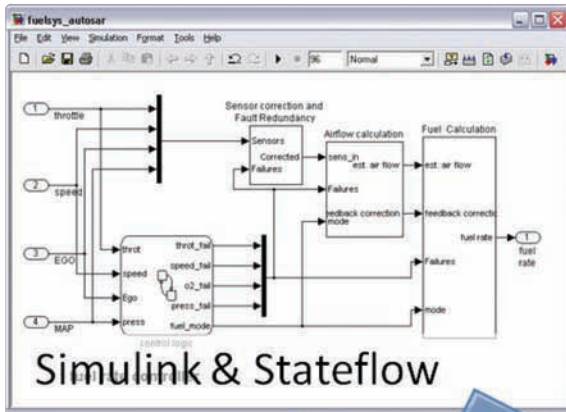
Por lo general, el ingeniero de diseño realiza mejoras en el modelo relacionadas con AUTOSAR, por ejemplo haciendo que los subsistemas se puedan llamar y, en consecuencia, permitiendo que se conviertan en runnables independientes u objetos de interfaz. En este caso particular, el ingeniero tiene que crear la configuración correspondiente a fin de garantizar que el código generado es compatible con el estándar y encaja en el entorno de software adicional que contiene el entorno de tiempo de ejecución y los componentes relativos al hardware procedentes de la capa de software básico. Esta configuración se puede crear a través de los cuadros de diálogo de parámetros de configuración adecuados. Para la generación de código, es preciso seleccionar el correspondiente AUTOSAR System Target en Real-Time Workshop® Embedded Coder.



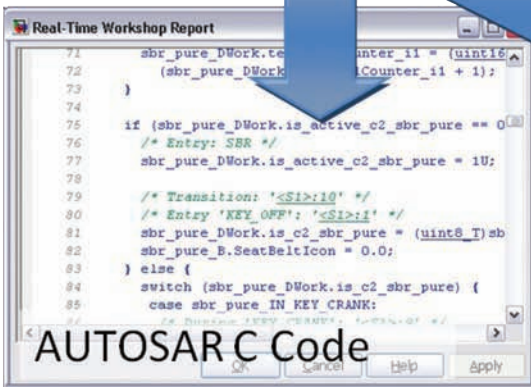
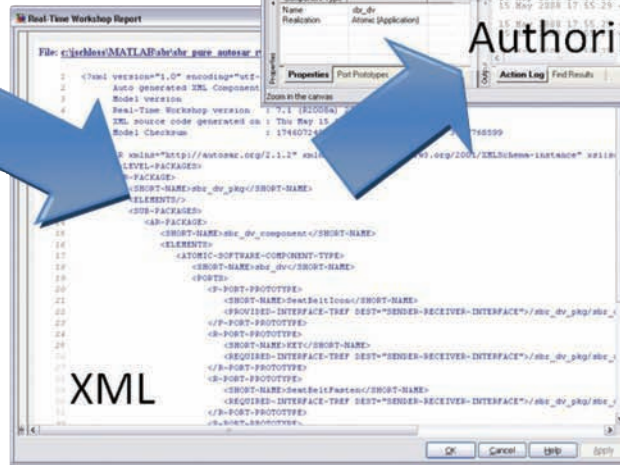
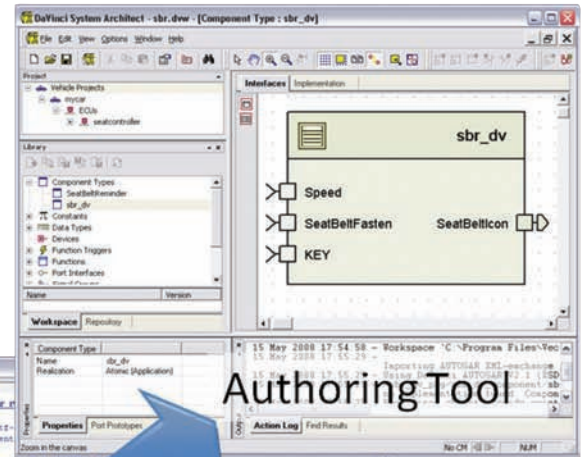
Esta segunda parte del flujo de trabajo es el mismo procedimiento que se utiliza en el siguiente flujo de trabajo.

Figura 3. Eportación de arquitecturas de DaVinci e importación en MATLAB y Simulink

Figura 4. Diseño de controladores (otro ejemplo)



arriba requiere las mismas configuraciones de AUTOSAR que se describen en el flujo de trabajo de arriba abajo. En concreto, los objetos de interfaz se configuran a fin de garantizar que el componen-



De abajo arriba: reutilización de modelos heredados para generar código compatible con AUTOSAR

Puesto que el diseño basado en modelos es un proceso ampliamente utilizado en el sector de la automoción, las compañías suelen tener una vasta biblioteca de modelos perfectamente definidos y ampliamente probados. Es importante que estos modelos se puedan reutilizar en distintas plataformas, como AUTOSAR, sin tener que realizar cambios en los bloques de los modelos.

Para hacer que un modelo existente sea compatible con AUTOSAR, siga este procedimiento:

1. Cambie el target en Real-Time Workshop Embedded Coder a "autosar".
2. Vaya a la pestaña AUTOSAR, abra el cuadro de diálogo de configuración de interfaz y seleccione la configuración predeterminada.
3. Realice los cambios oportunos para configurar los puertos de remitente y receptor distintos de los predeterminados y seleccione "Validar".
4. Inicie la generación de código.

El flujo de trabajo de abajo

de software generado se puede integrar correctamente. Además de los archivos de código, también se generará de forma automática un archivo de descripción de componente de software de AUTOSAR en formato XML, tal y como se muestra en la Figura 4.

Y ahora, ¿qué es lo siguiente?

Ha quedado claro que AUTOSAR no es tan aterrador como parece en ocasiones. Aunque los arquitectos de sistemas necesitan un conocimiento más profundo de "los entresijos de AUTOSAR", los ingenieros de desarrollo de funciones no deben preocuparse de ese nivel de detalle.

Claro está que podemos pensar en capacidades y escenarios adicionales. Las próximas versiones de AUTOSAR demostrarán la capacidad del estándar para solucionar los problemas de complejidad de este sector. Al fin y al cabo, los ingenieros deben ser capaces de superar sus desafíos, mientras que los proveedores de herramientas deben apoyarlos lo máximo posible.

Referencias

- Fürst, S. (2008). AUTOSAR – An open standardized software architecture for the automotive industry. 1st AUTOSAR Open Conference & 8th AUTOSAR Premium Member Conference. Detroit, MI, USA.
- Köhler, A., & Reck, T. (2007). AUTOSAR-Compliant Functional Modeling with MATLAB®, Simulink®, Stateflow® and Real-Time Workshop® Embedded Coder of a Serial Comfort Body Controller. MathWorks Automotive Conference. Dearborn, MI, USA.
- Lajtkep, A. (2008). BMW's strategy to introduce AUTOSAR. Vector Congress 2008. Stuttgart.
- Nesci W., A. S. (2008). Magneti Marelli System and Application development with AUTOSAR. 1st AUTOSAR Open Conference & 8th Premium Member Conference. Detroit, MI, USA.
- Sandmann, G., Schlosser, J., & Thompson, R. (2008). Development of AUTOSAR Software Components. International Automotive Conference. Stuttgart: The MathWorks.

Figura 5. Generación de código, exportación de XML e importación a DaVinci